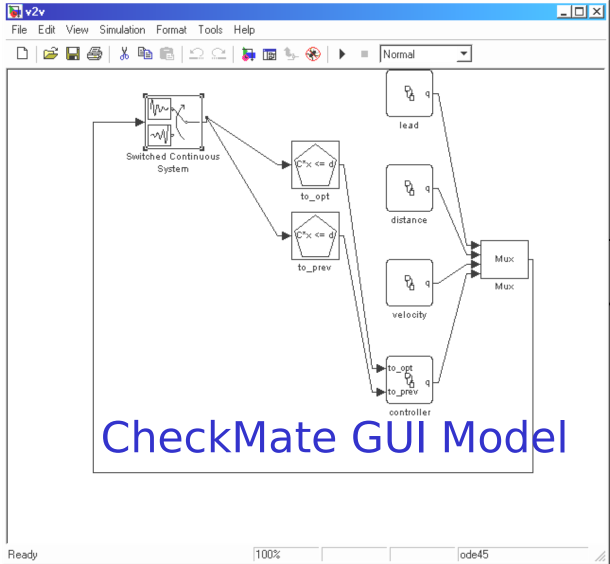
New Developments:

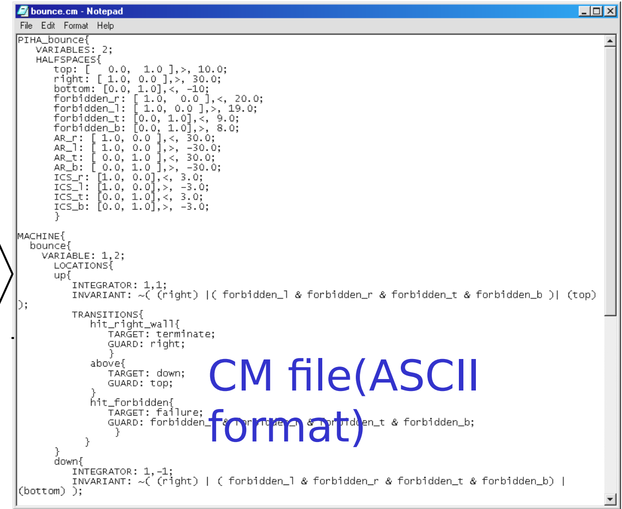
- *CM modeling language* - interface to HSIF
- *oriented rectangular hull* - efficient alternative to the convex hull
- *verification of the ETC model* - nonlinear dynamics with uncertain parameters
- *sampled-data verification – verification of mode-switching real-time controllers*
- *counter-example-guided refinement – extension of discrete refinement techniques to hybrid systems*

Approved for Public Release, Distribution Unlimited

CheckMate←→HSIF



CheckMate GUI Model



```
PIHA_bounce{
   VARIABLES: 2;
   HALFSPACES{
      top: [   0.0,  1.0 ],>, 10.0;
      right: [ 1.0, 0.0 ],>, 30.0;
      bottom: [0.0, 1.0],<, -10;
      forbidden_r: [ 1.0,  0.0 ],<, 20.0;
      forbidden_l: [ 1.0, 0.0 ],>, 19.0;
      forbidden_t: [0.0, 1.0],<, 9.0;
      forbidden_b: [0.0, 1.0],>, 8.0;
      AR_r: [ 1.0, 0.0 ],<, 30.0;
      AR_l: [ 1.0, 0.0 ],>, -30.0;
      AR_t: [ 0.0, 1.0 ],<, 30.0;
      AR_b: [ 0.0, 1.0 ],>, -30.0;
      ICS_r: [1.0, 0.0],<, 3.0;
      ICS_l: [1.0, 0.0],>, -3.0;
      ICS_t: [0.0, 1.0],<, 3.0;
      ICS_b: [0.0, 1.0],>, -3.0;
      }

MACHINE{
  bounce{
    VARIABLE: 1,2;
      LOCATIONS{
      up{
         INTEGRATOR: 1,1;
         INVARIANT: ~( (right) |( forbidden_l & forbidden_r & forbidden_t & forbidden_b )| (top)
);
         TRANSITIONS{
            hit_right_wall{
               TARGET: terminate;
               GUARD: right;
               }
            above{
               TARGET: down;
               GUARD: top;
            }
            hit_forbidden{
               TARGET: failure;
               GUARD: forbidden_l & forbidden_r & forbidden_t & forbidden_b;
                }
           }
      }
      down{
         INTEGRATOR: 1,-1;
         INVARIANT: ~( (right) | ( forbidden_l & forbidden_r & forbidden_t & forbidden_b) |
(bottom) );
```

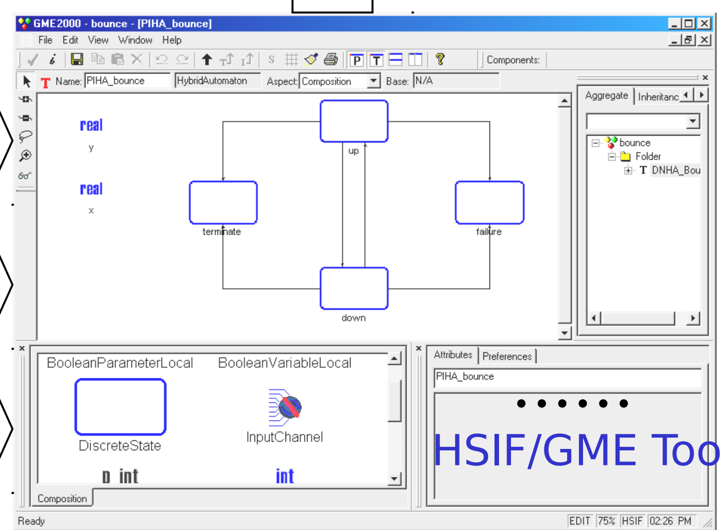
CM file(ASCII format)

- Multiple concurrent hybrid automata
- Linear/Nonlinear dynamics
- Guard/Invariant defined as boolean combination of half spaces
- Continuous state space globally defined

Charon

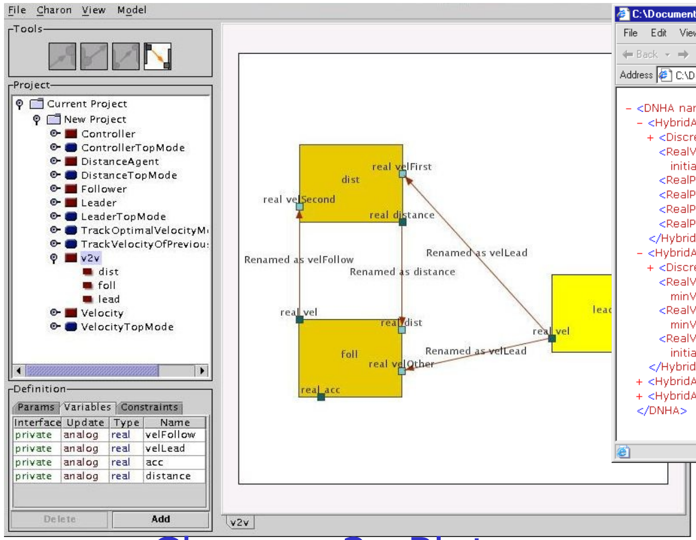
Ptolemy

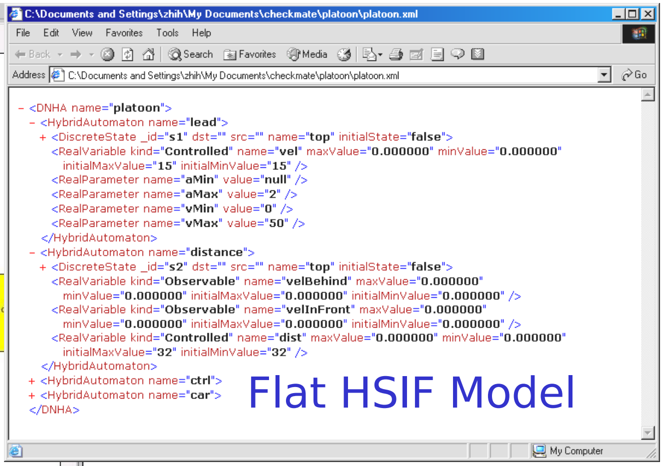
SAL



HSIF/GME Tool

V2V HSIF → CheckMate

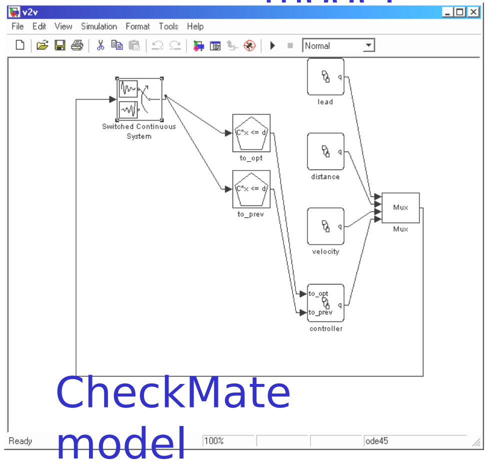




Flat HSIF Model

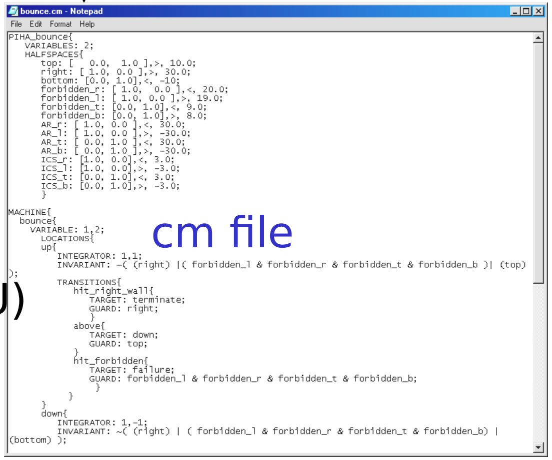
Charon v2v Platoon model

GME/CM Interpreter (Vanderbilt)



CheckMate model

CM2CheckMate Converter (CMU)

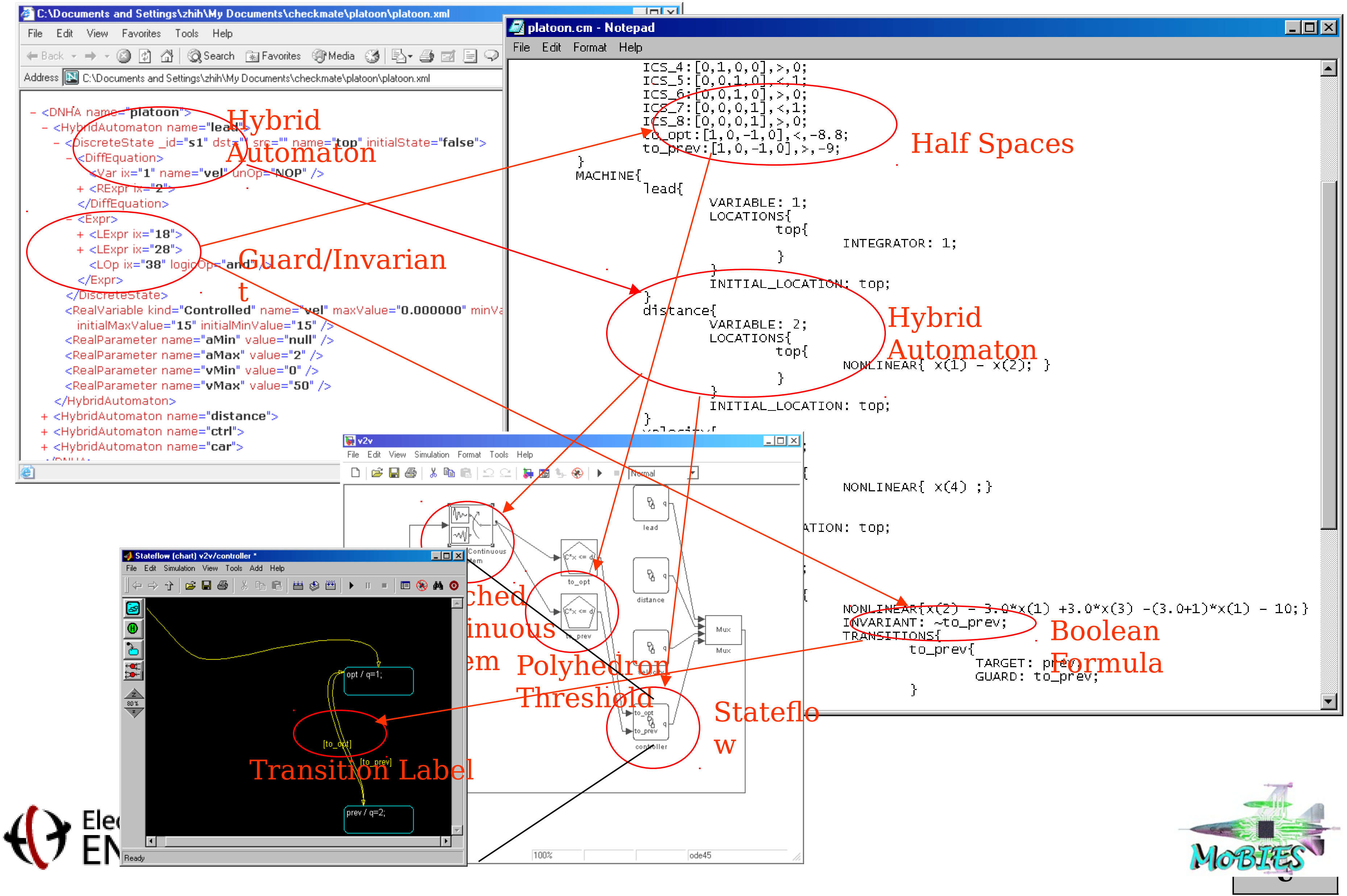


cm file

Electrical & Computer ENGINEERING

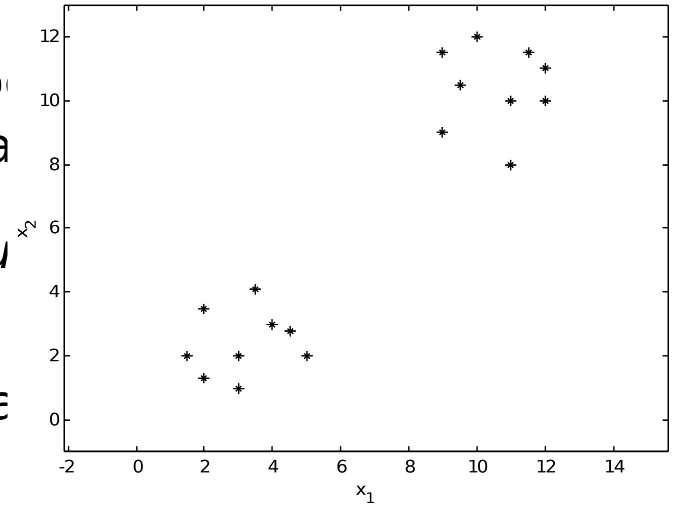
MoBIES

V2V HSIF→CheckMate



Set Representation by Oriented Rectangular Hulls

Task:
determine a hull that contains a given set of p
(required, e.g., during the flow pipe approxima



Solution so far: computation of a *convex hu*

Drawbacks: - time consuming

- high and increasing number of fa

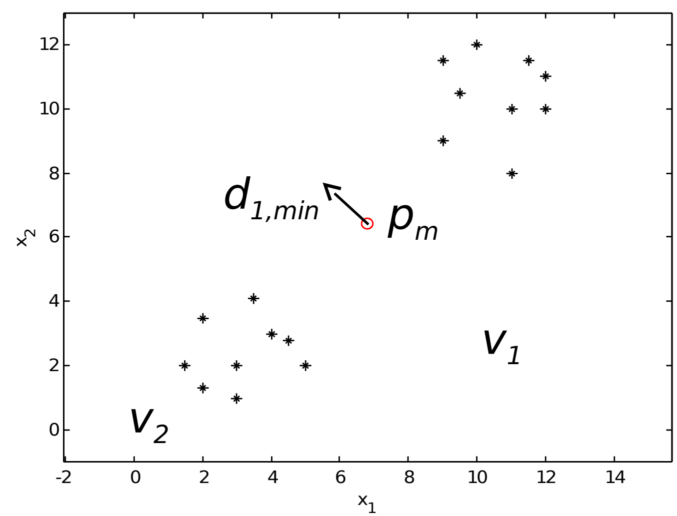
Computation of Oriented Rectangular Hulls:

set of points: $P = \{p_1, \ldots, p_q\}$

new coordinate system:

- origin: $p_m = \sum_{i=1}^{q} p_i / q \quad \Rightarrow \quad P' = \{p' \mid p \in P : p' = p - p_m\}$

- orientation: directions v_1, v_2 from singular value decomposition of $P'(P')^T$



min/max values over P' in the new directions

determine the boundaries of the rectangular hull

Comparison of Convex and Oriented Rectangular Hulls

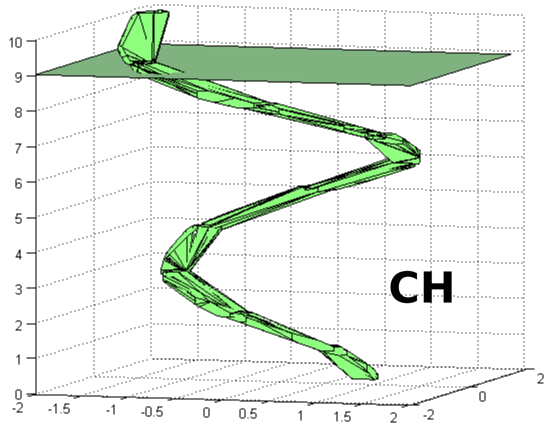
Approximation accuracy / number of faces:

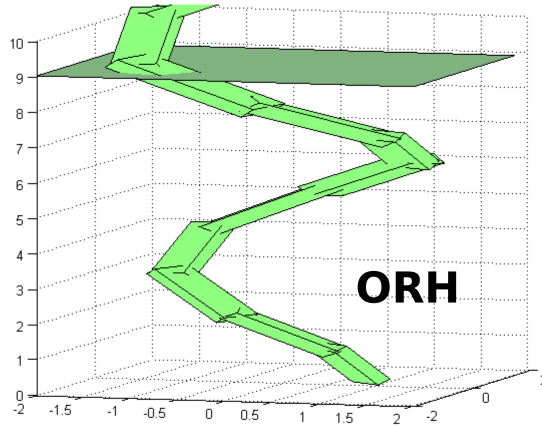
van-der Pole system:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{x_2}{-5}\cdot\left(x_1^2 - 1\right) - x_1$$

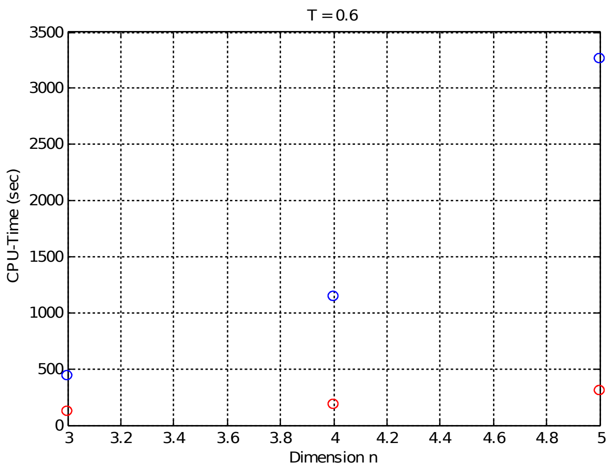
$$\dot{x}_3 = 1$$





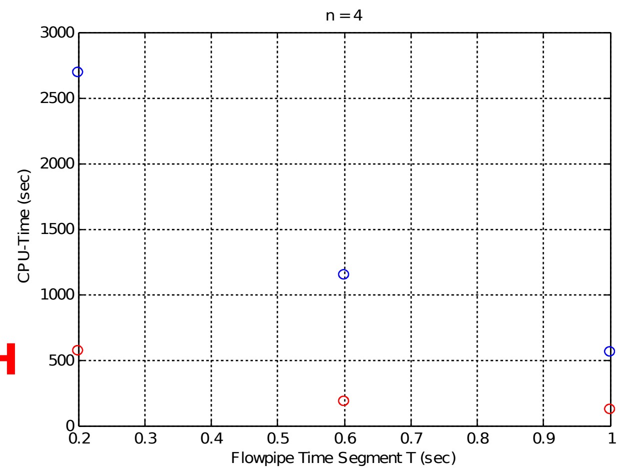
ORH:
→ slightly worse approximation
→ considerably less faces of the polyhedra

Computational complexity (same example):



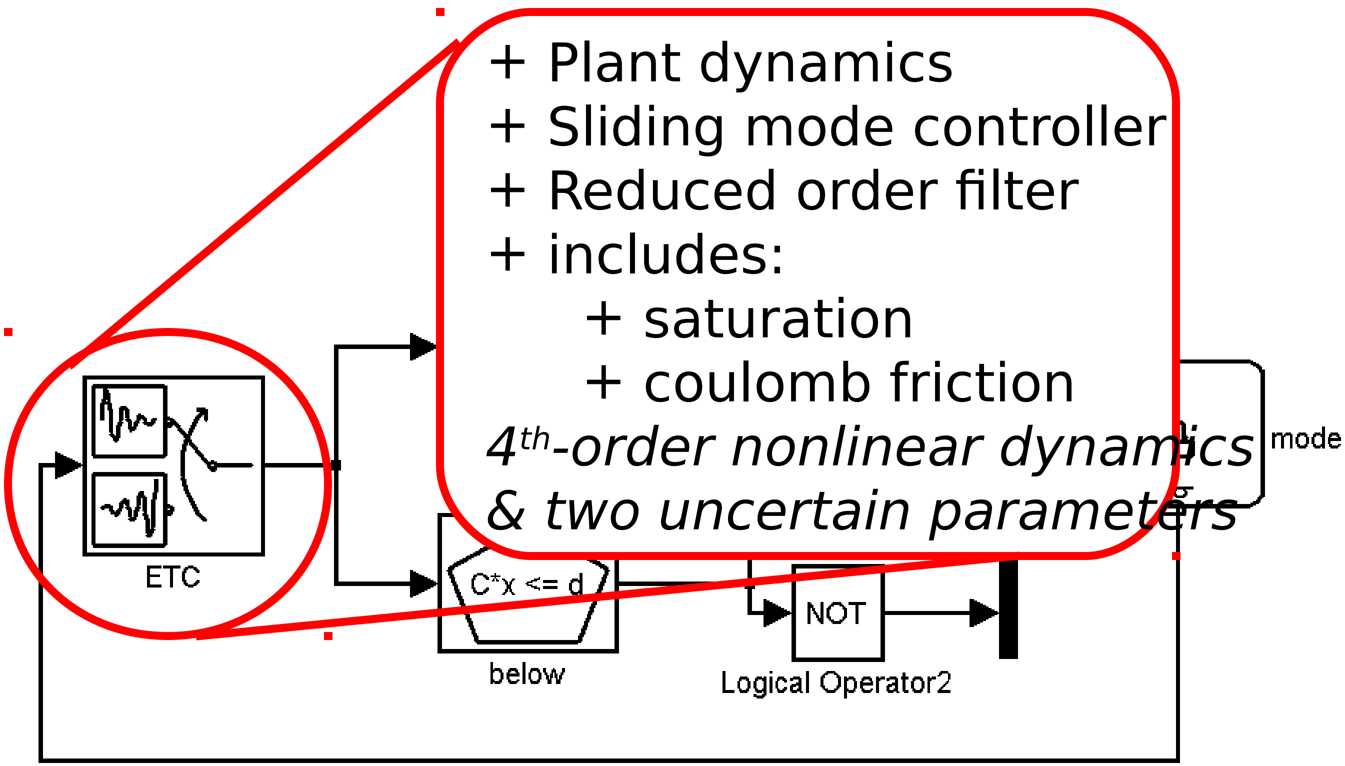
CH

ORH



ORH:
significantly smaller computation times in all cases

ETC CheckMate Model



+ Plant dynamics
+ Sliding mode controller
+ Reduced order filter
+ includes:
    + saturation
    + coulomb friction

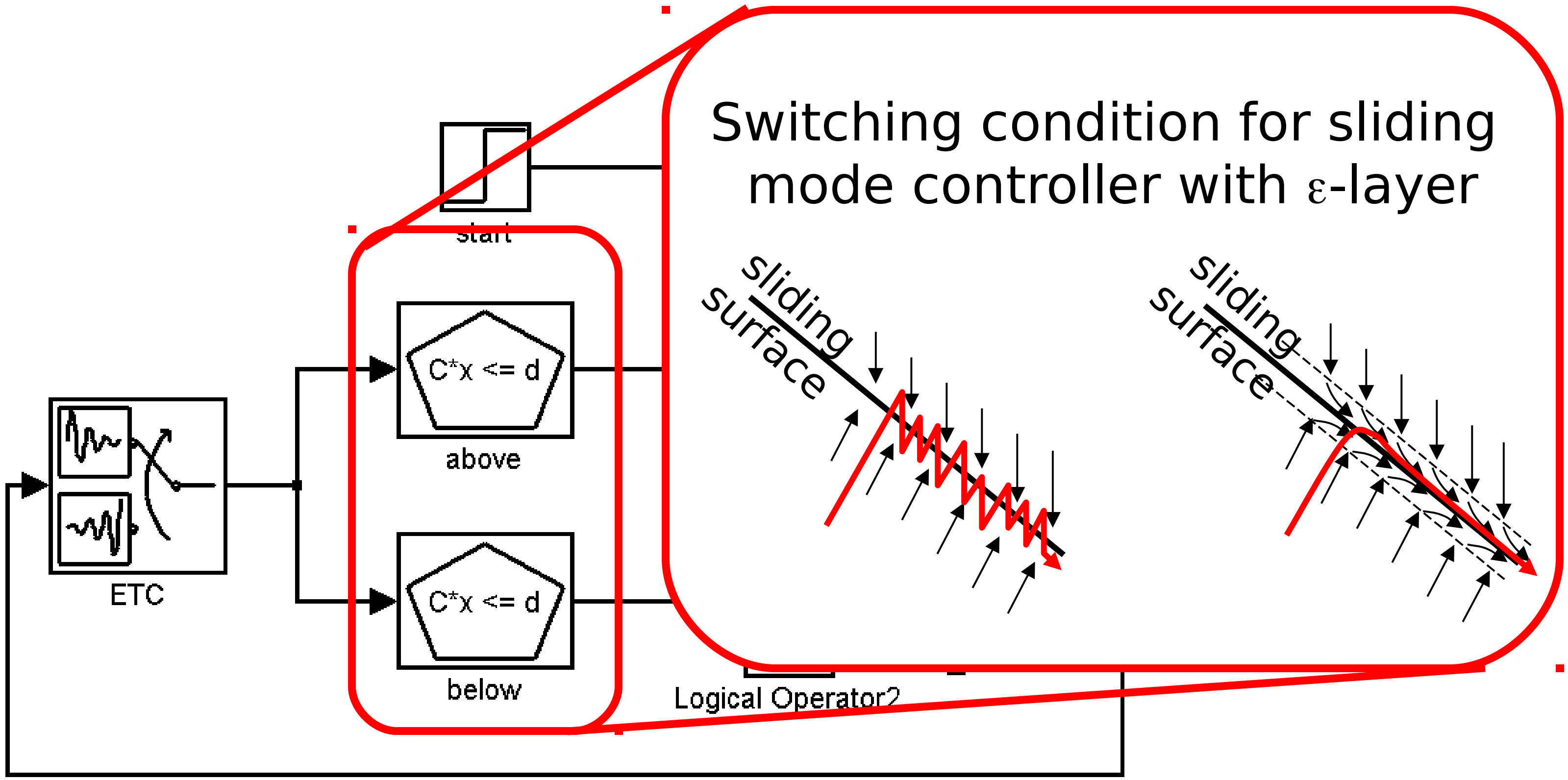
4th-order nonlinear dynamics & two uncertain parameters

Decomposition in
- switched continuous system
- switching conditions
- switching logic

Electrical & Computer ENGINEERING

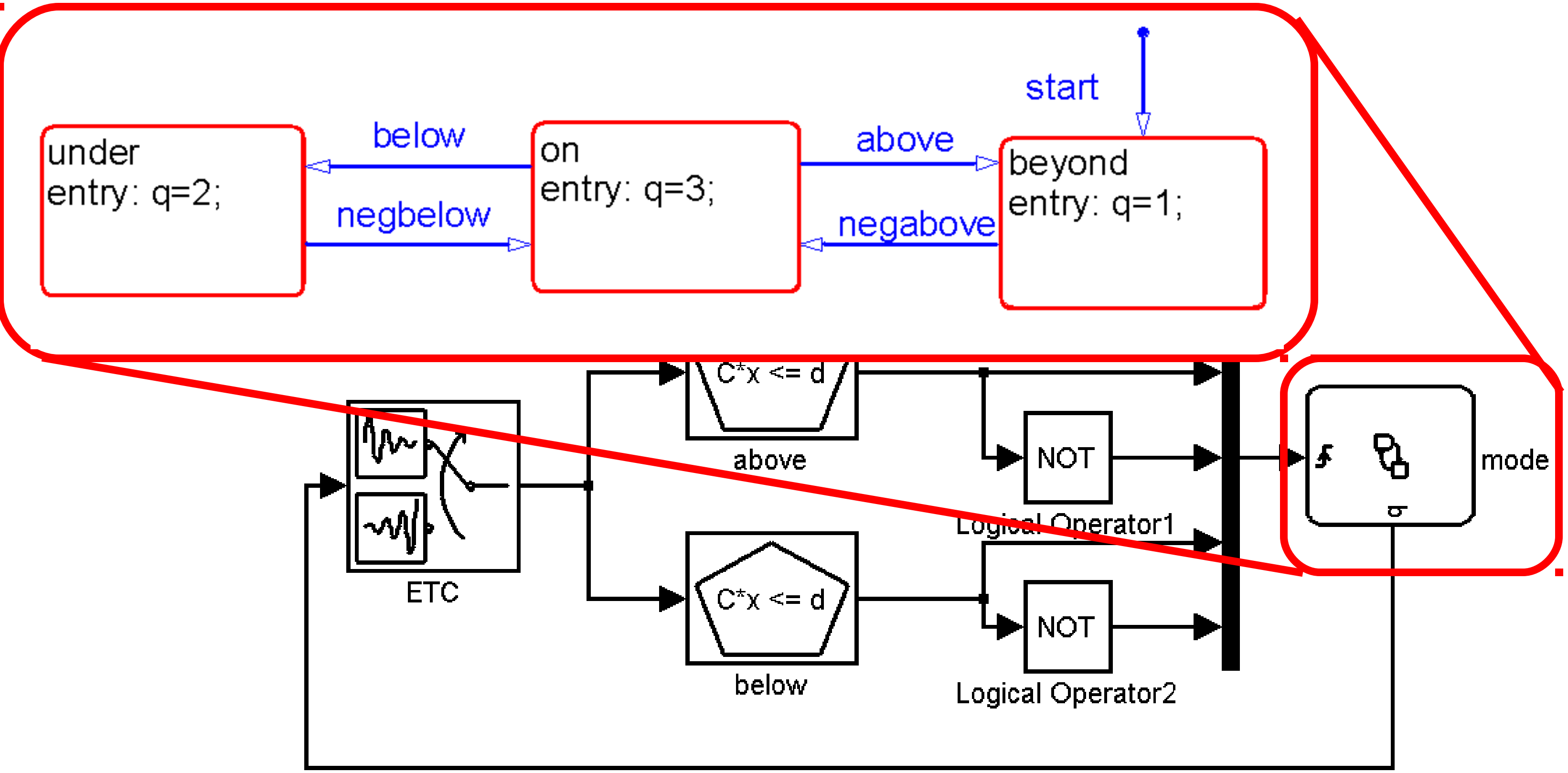
MoBIES

ETC CheckMate Model



Decomposition in
- switched continuous system
- switching conditions
- switching logic

ETC CheckMate Model



Decomposition in

- switched continuous system
- switching conditions
- switching logic

Parametric Verification

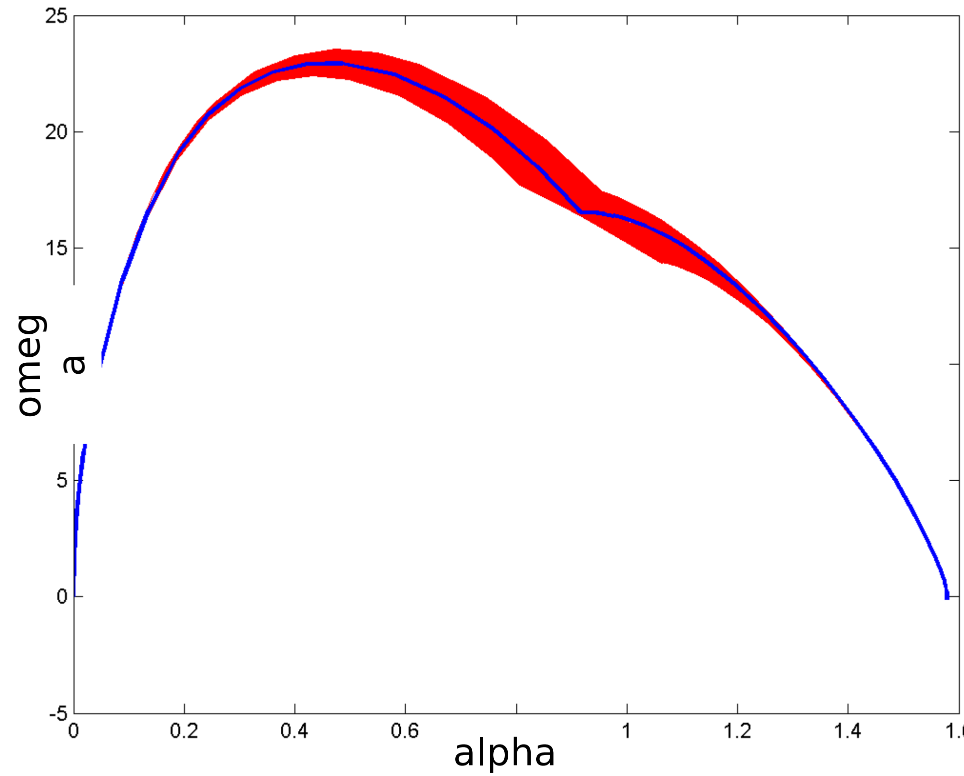
Most ETC requirements are formulated for a single trajectory

Implicit assumption:

- Controller has exact parameter values

Simulation of the vertices not sufficient!

±20%



Parametric verification explores behavior for all possible parameter v

Verification Results

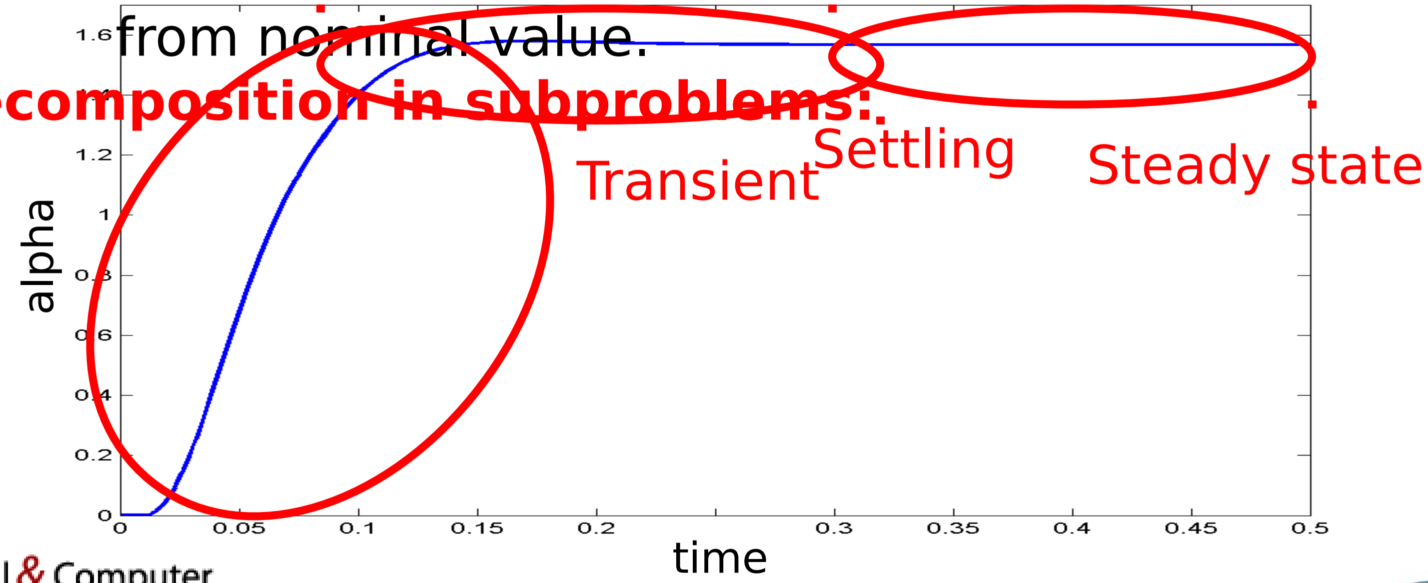
Steady state tracking error:

Difference between desired angle and actual angle <2 %

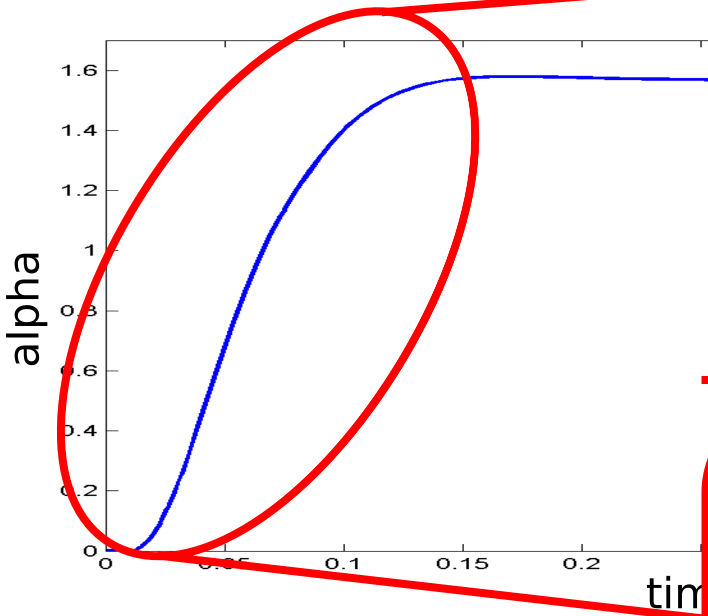
Uncertain parameters:

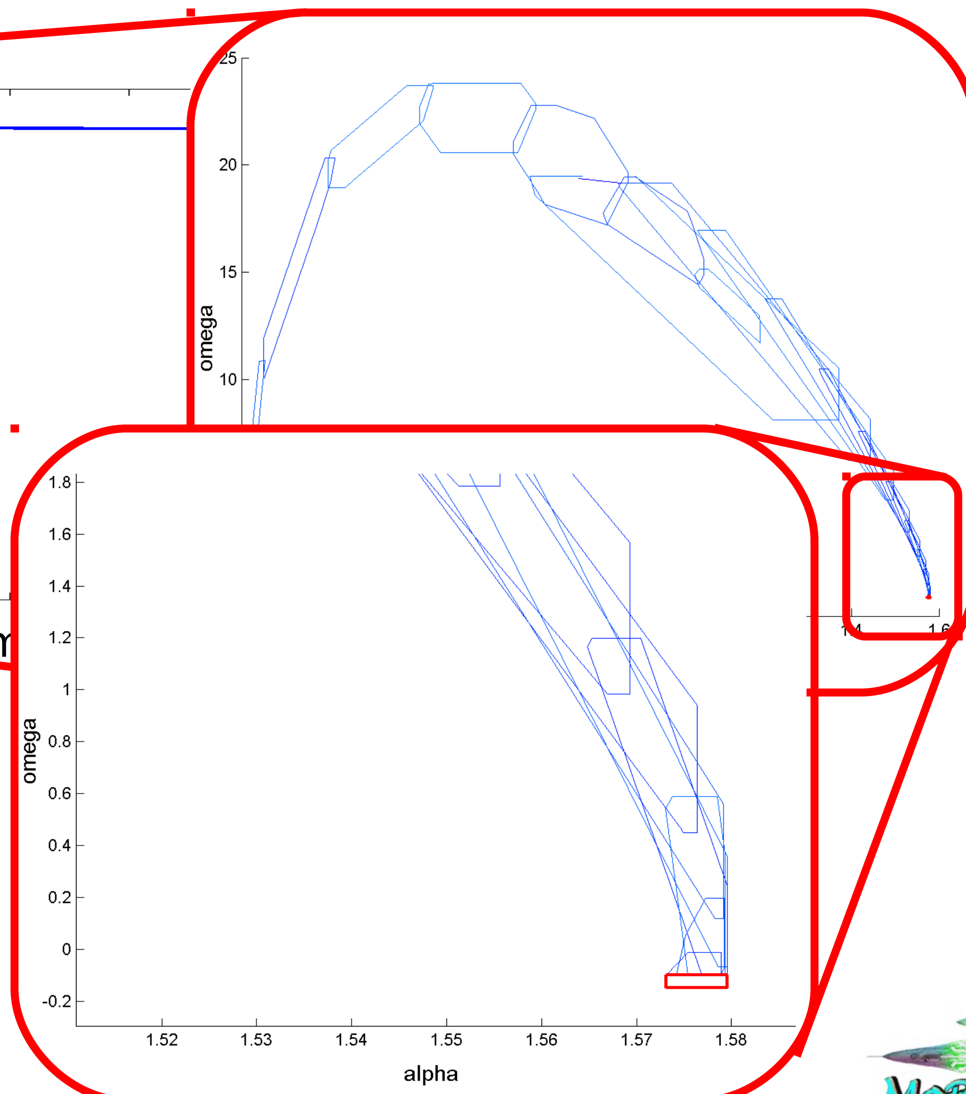
Spring constant and *spring equilibrium* may deviate by 20% from nominal value.

Decomposition in subproblems:



Verification Results





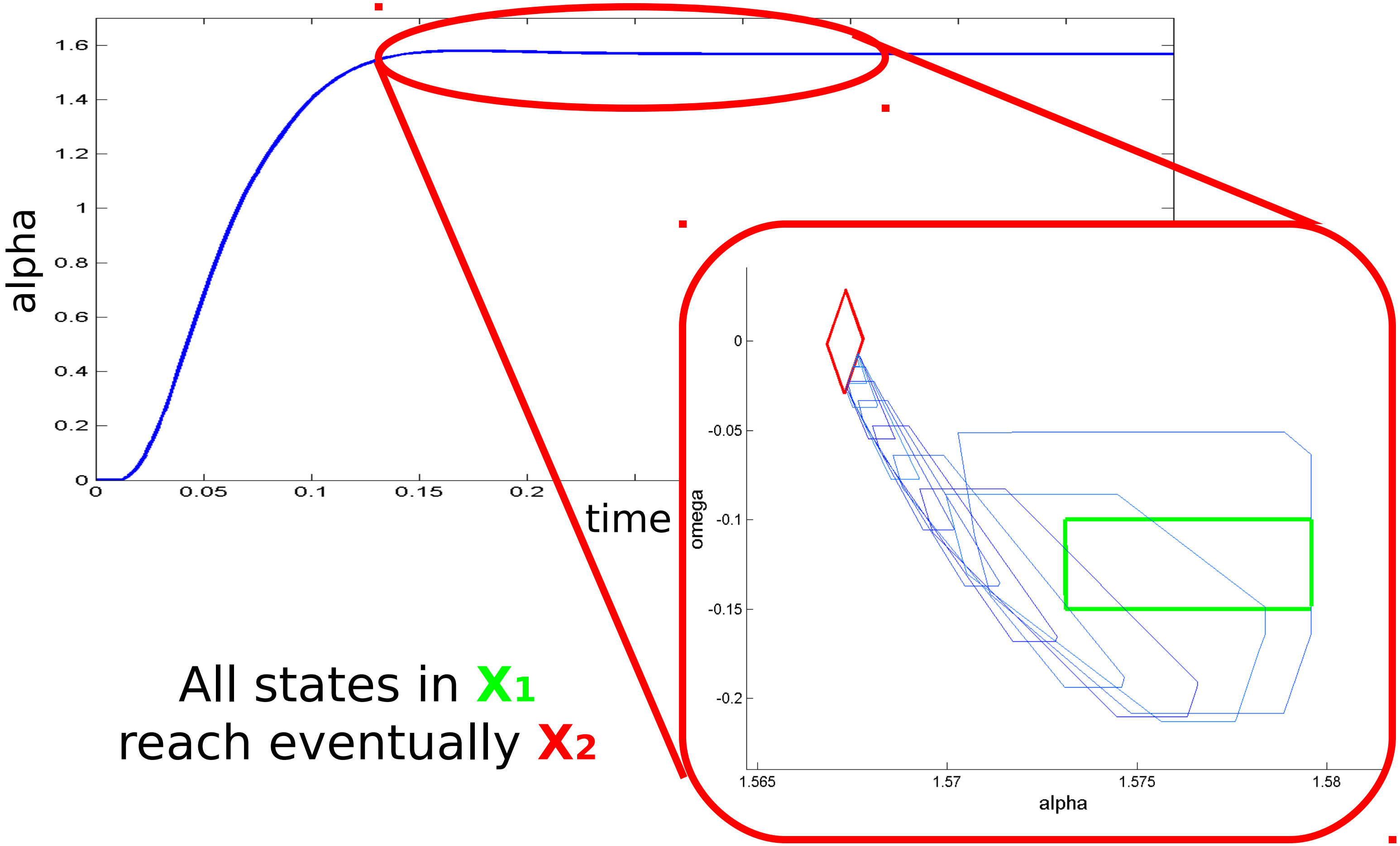
All states in X_0
reach eventually X_1

Electrical & Computer ENGINEERING

~30 min. on 1.5 GHz Pentium 4

MoBIES

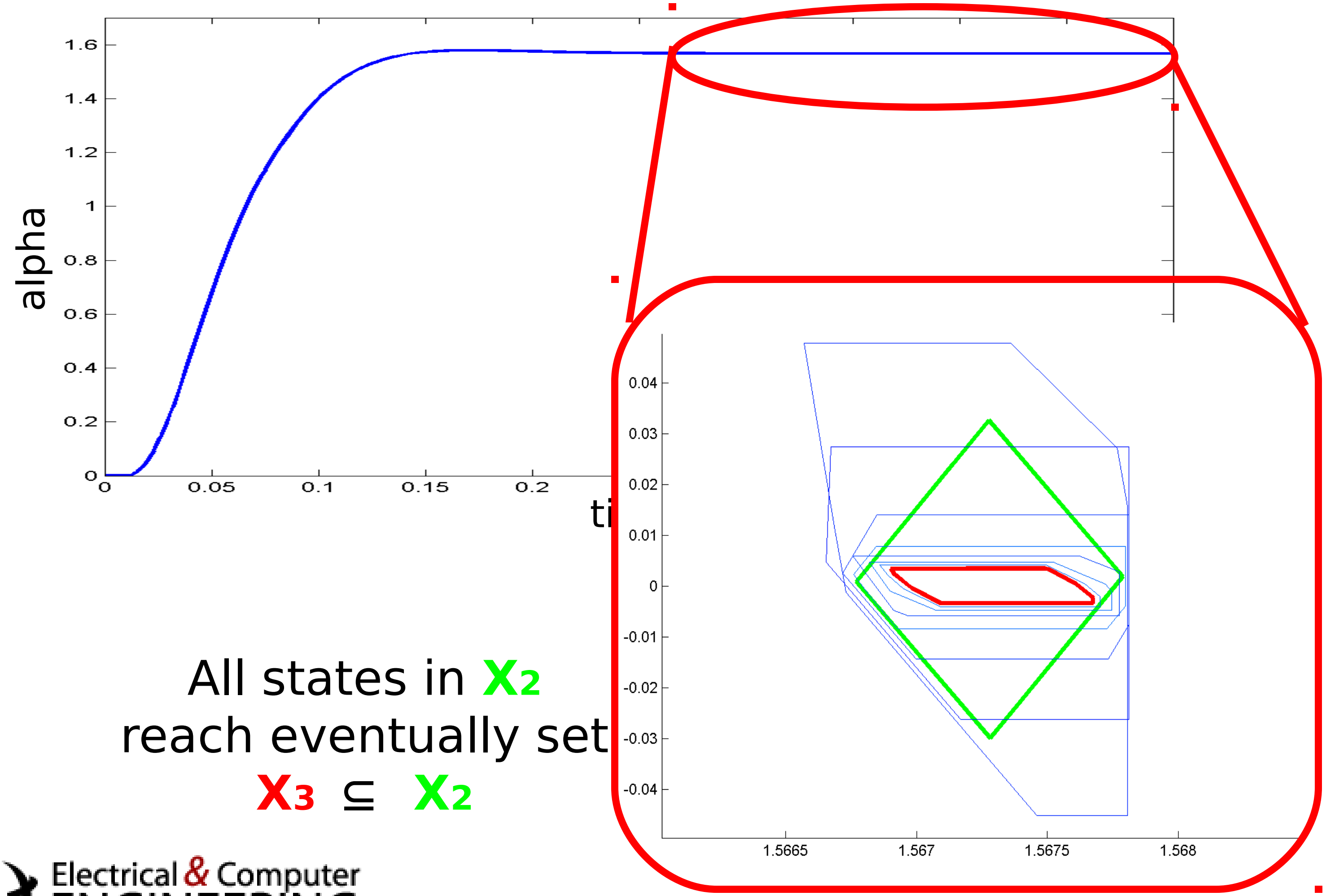
Verification Results



All states in X_1
reach eventually X_2

~22 min. on 1.5 GHz Pentium 4

Verification Results



All states in X_2 reach eventually set $X_3 \subseteq X_2$

~19 min. 1.5 GHz Pentium 4

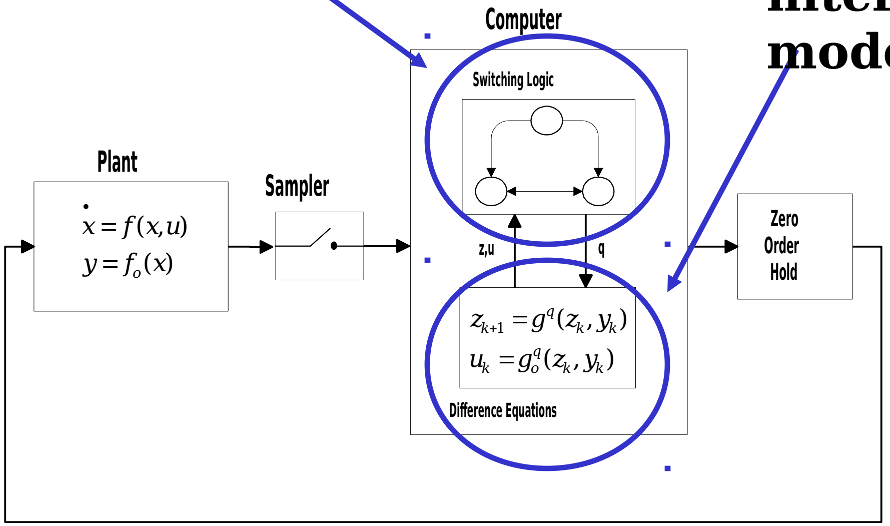
Electrical & Computer ENGINEERING

MoBIES

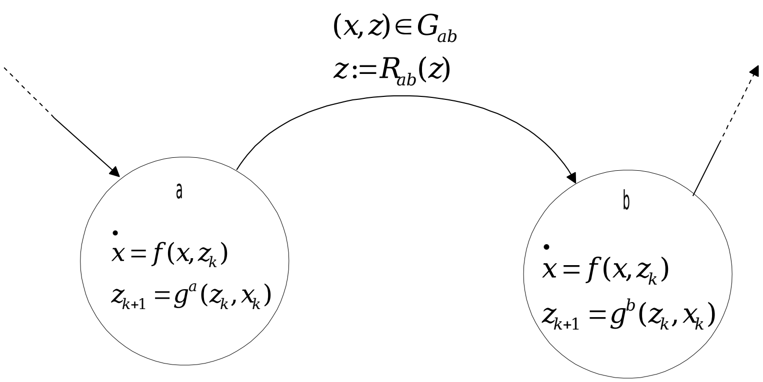
Multi-Mode Embedded Control Systems

Mode Switching Logic

Floating Point Computations (e.g. PID, filters, sliding mode control)



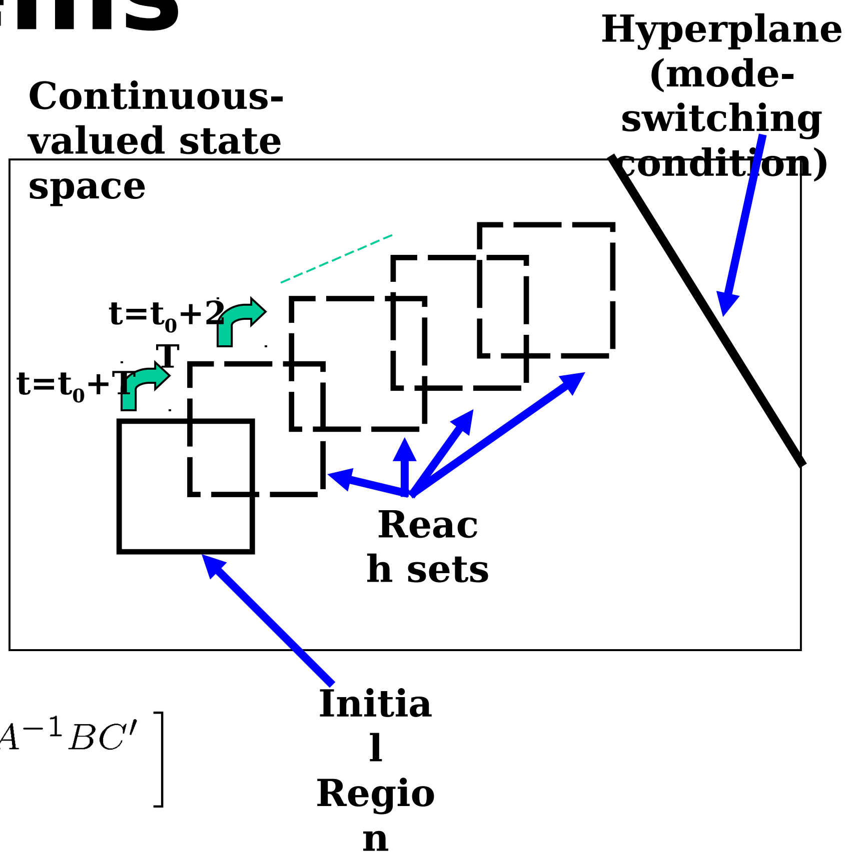
Switched-mode Control



Equivalent Sample-Data Hybrid Automaton

J. Kapinski, B. Krogh, "Verifying Switched-Mode Computer Controlled Systems," *Computer Aided Control System Design Conference*, Sept. 2002

Computing Reachable Sets for Sampled-Data Hybrid Systems



When the plant/controller are affine

$$
\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(k) + r \\
y(t) &= Cx(t) \\
z(k+1) &= A'z(k) + B'y(k) + r' \\
u(k) &= C'z(k) + D'y(k) + r''
\end{aligned}
$$

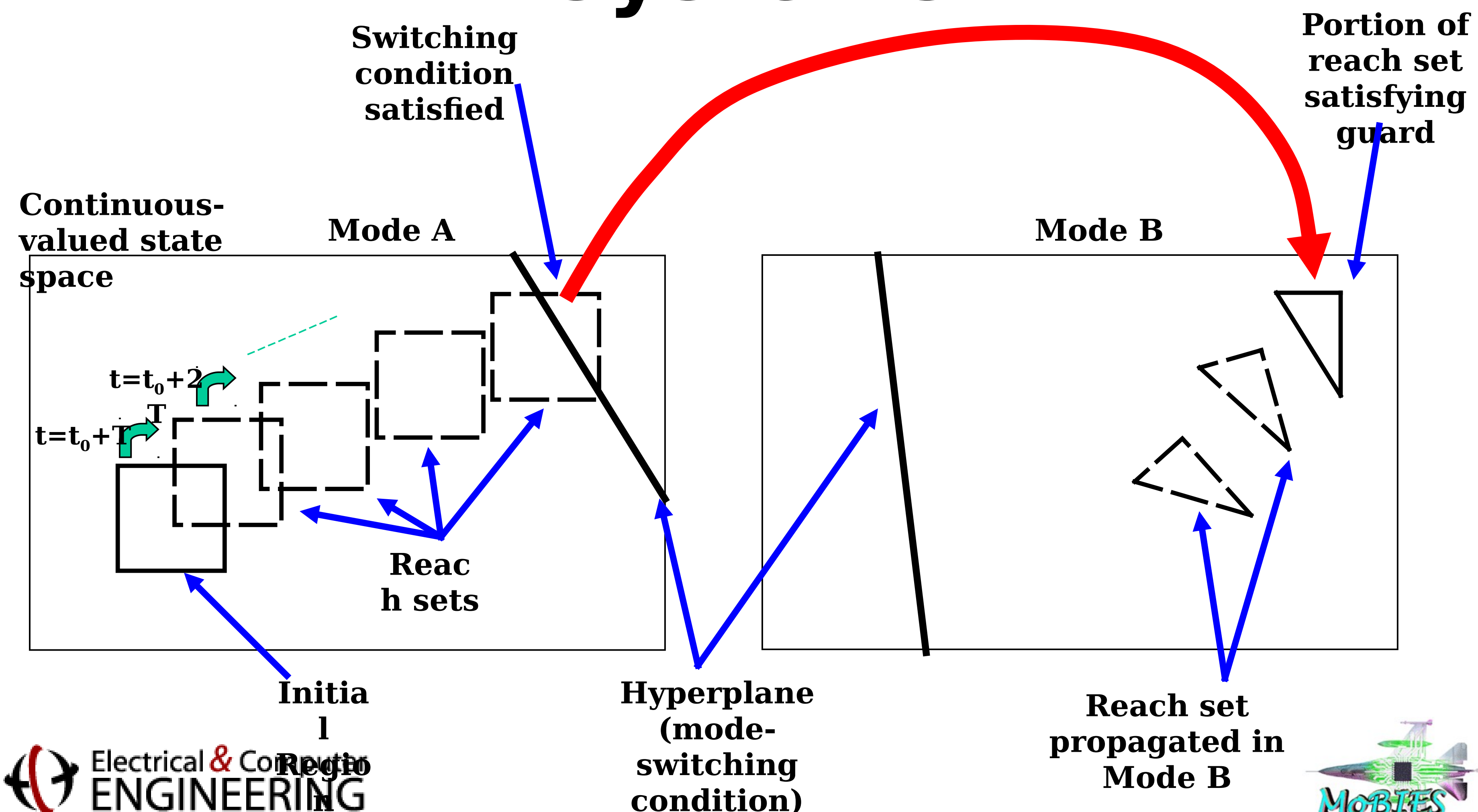
The reachset is given by the affine transformation

$$
\begin{aligned}
P &= \begin{bmatrix} e^{(AT)} + (e^{(AT)} - I)A^{-1}BD'C & (e^{(AT)} - I)A^{-1}BC' \\ B'C & A' \end{bmatrix} \\
v &= \begin{bmatrix} (e^{(AT)} - I)A^{-1}(Br'' + r) \\ r' \end{bmatrix}
\end{aligned}
$$

Electrical & Computer ENGINEERING

MoBIES

Computing Reachable Sets for Sampled-Data Hybrid Systems



Variable Cam Timing (VCT) Example

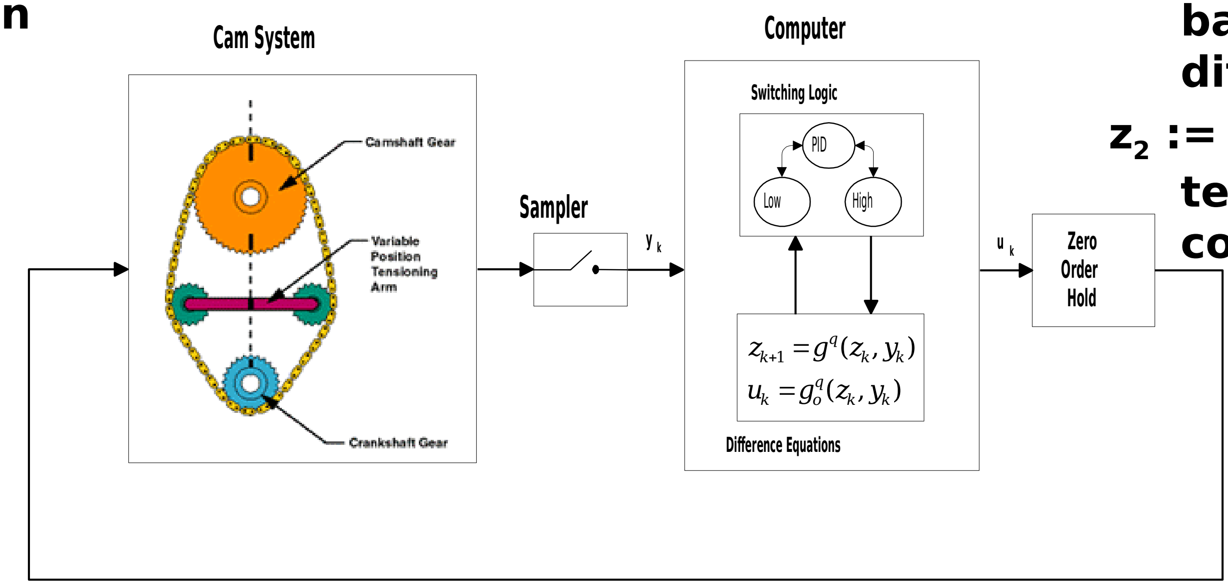
Plant state variable:

x := actuator position

Controller state variables:

z_1 := used for band-limited differentiator

z_2 := integrator term for PID control



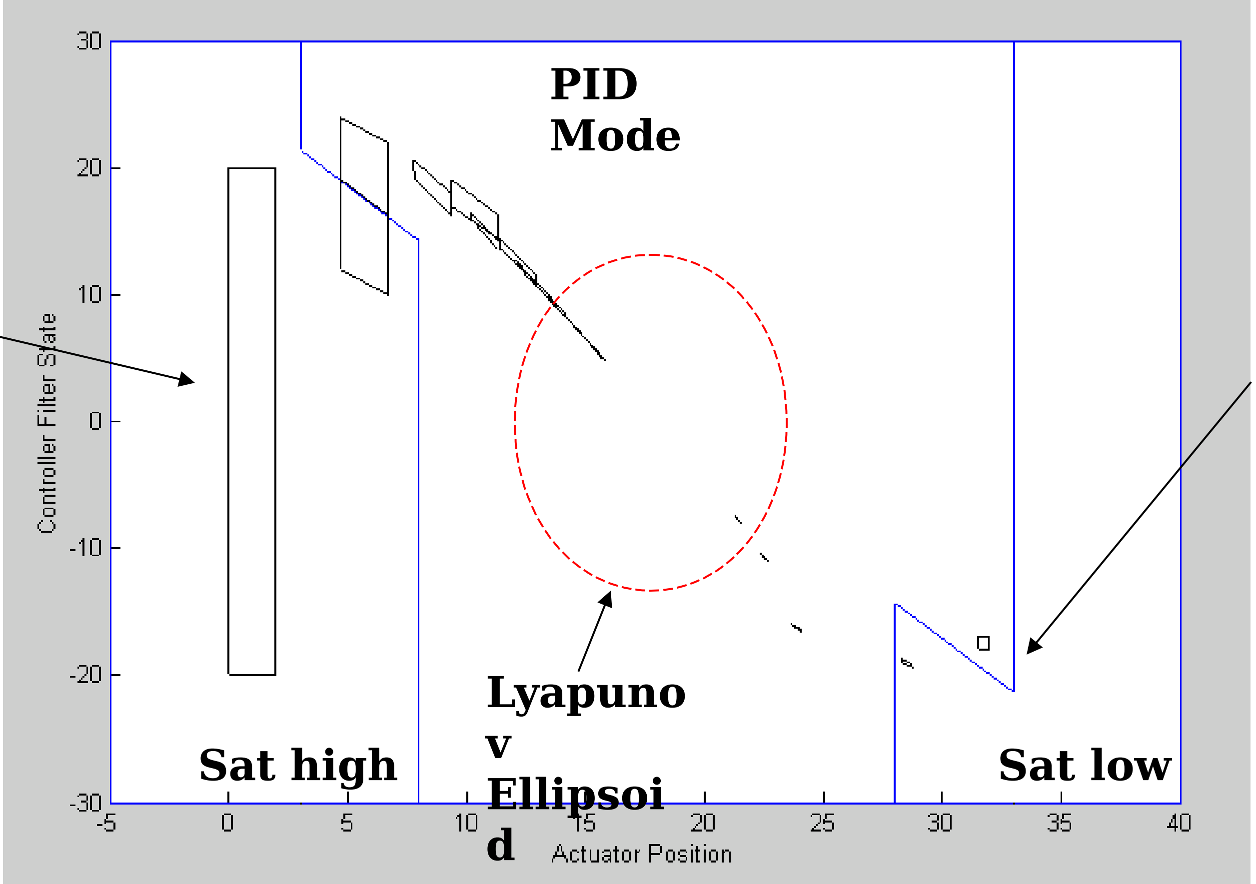
- Requirement: system should not switch modes more than once

VCT Sampled-Data Verification

Safe ICS

Verification step:

80.6 seconds



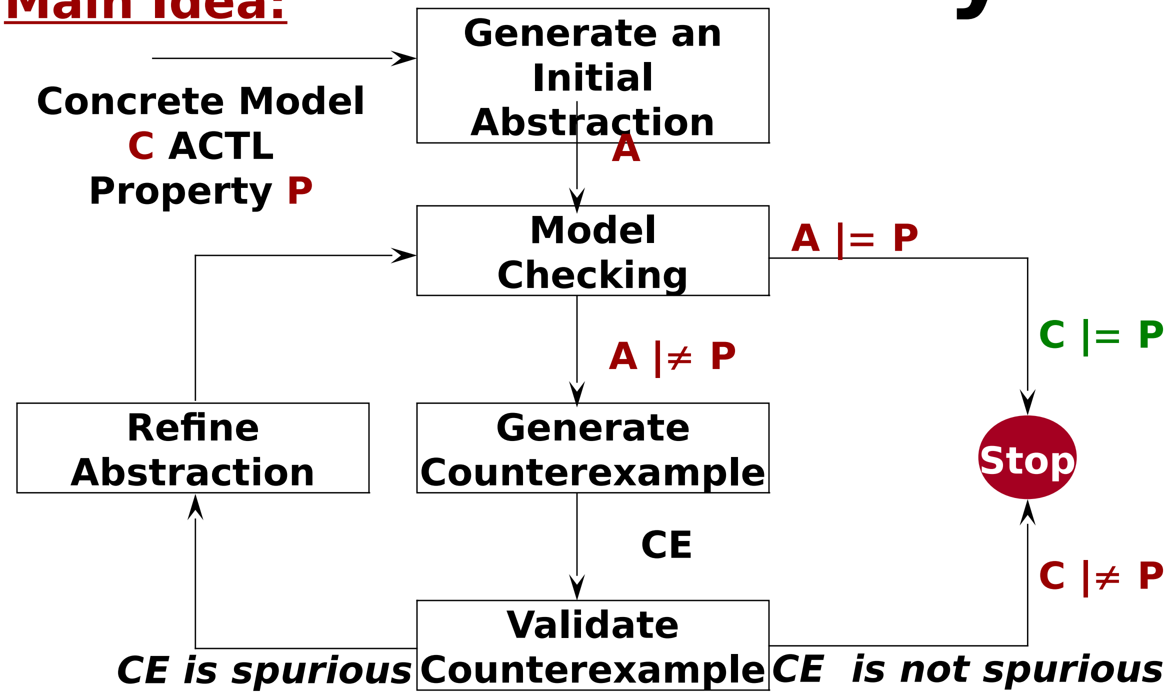
Unsafe ICS

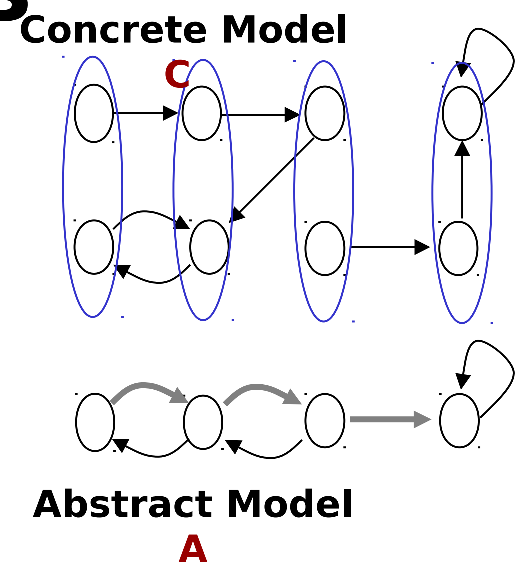
Verification step:

41.7 seconds

Counterexample-Guided Abstraction Refinement for Finite-State Systems

Main Idea:





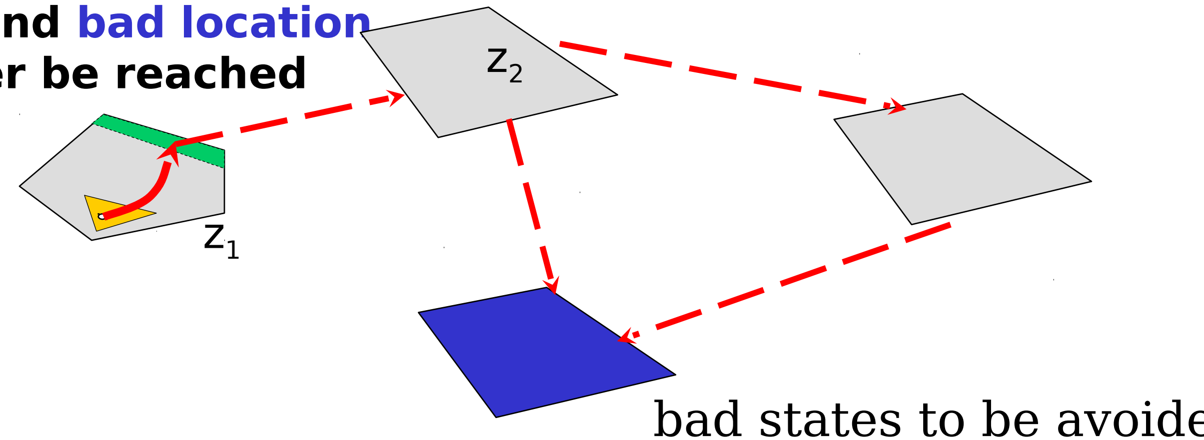
Previous Work:

- R. Alur, T. Dang, F. Ivancic: *Reachability Analysis of Hybrid Systems via Predicate Abstraction*. HSCC-2002.
- E. Clarke, O. Grumberg, S. Jha, Y. Lu, H. Veith: *Counterexample-guided Abstraction Refinement*. CAV-2000.
- E. Clarke, A. Gupta, J. Kukula, O. Strichman: *SAT-based Abstraction-Refinement using ILP and Mach.-Learning* CAV-2002.
- T. Henzinger, R. Jhala, R. Majumdar, G. Sutre: *Lazy Abstraction*. POPL-2002.
- K.S. Namjoshi, R.P. Kurshan: *Syntactic Program Transformations for Automatic Abstraction*. CAV-2000.

Counterexample-Guided Abstraction Refinement for Hybrid Systems

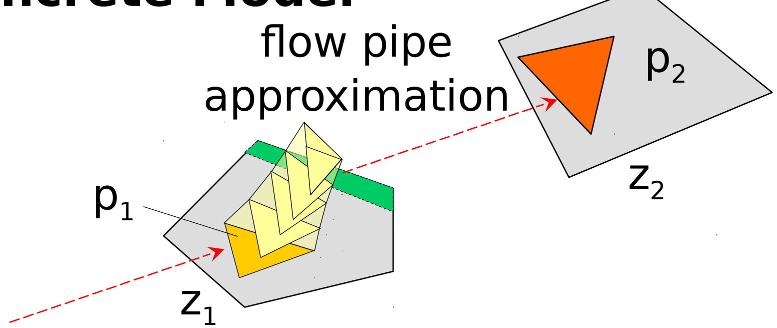
Verification Problem:

Given: initial location + set and bad location
Verify: Bad location can never be reached

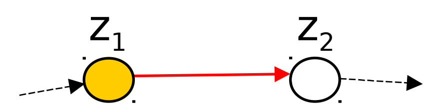


Abstraction Refinement:

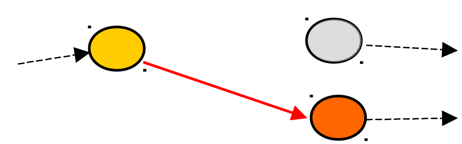
Concrete Model



Abstract Model:



Refined Abstract Model:



Implementation of Counterexample- Guided Abstraction Refinement

- The new approach is currently being implemented and incorporated into CheckMate
- Initial prototype used to validate correctness of approach
  - Applied to the verification of a simple car controller
  - Efficiency limited by direct usage of CheckMate routines/data structures
- Improved prototype to be completed shortly
  - Based on new data structures tailored to counter-example-guided refinement
  - Supports different algorithms to validate counterexamples

<u>Technical Report:</u>

E. Clarke, A. Fehnker, Z. Han, B. Krogh, O. Stursberg, M. Theobald

Verification of Hybrid Systems Based on Abstraction and Counterexample-Guided Model Refinement (available shortly).

Electrical & Computer ENGINEERING

MoBIES

Next Steps

- complete integration & demonstration of CheckMate interface to HSIF
- extend demonstrations to OEP power train models
- implement and demonstrate nonlinear sampled-data verification
- complete comprehensive guidelines for performing hybrid system verification of embedded systems